基于改进蚁群算法的无人飞行器航迹规划*

熊 瑜 饶跃东
（桂林空军学院 桂林 541003）

摘 要 无人飞行器航迹规划是现代战争中实施远程精确打击，提高飞行器实际作战效能的关键技术。蚁群算法作为一种启发式仿生优化算法，能够有效应用于航迹规划中。针对基本蚁群算法在应用中容易过早陷入局部最优解这一缺点，提出自适应动态双种群蚁群算法的改进策略，通过信息素的震荡变化和挥发系数的自适应调整，扩大搜索空间，提高算法搜索的全局性。并将改进后的算法应用于无人飞行器航迹规划，通过实验仿真，证明了此改进算法在航迹规划应用中的可行性和有效性。

关键词 航迹规划；蚁群算法；自适应；信息素
中图分类号 V249

Route Planning of UAV Based on Improved Ant Colony Algorithm

Xiong Yu Rao Yuedong
(Guilin Air Force Academy, Guilin 541003)

Abstract Route planning of UAV is an important technique in long-range precision strikes and improve the combat effectiveness of aircraft. Ant colony algorithm as a heuristic bionic optimization algorithm can be effectively applied to route planning. The prominent shortcoming of the basic ant colony algorithm is easily trapped into local optimal solution. Adaptive dynamic dual population ant colony algorithm is proposed in this paper in order to solve this problem. The concussion change of the pheromone and the adaptive adjustments of the volatile coefficient can expand the search space and improve the overall searching performance. It is proved that the algorithm is feasible and effective in the route planning simulation.

Key Words route planning, ant colony algorithm, adaptive, pheromone
Class Number V249

1 引言

无人飞行器（UAV，Unmanned Aerial Vehicle）航迹规划主要是指在综合考虑无人飞行器机动性能、环境地形、油耗代价、威胁代价等诸多特定约束条件下，在出发地和目的地之间寻找到一条满足某种性能指标的最优或最满意的飞行轨迹。在现代战争中，航迹规划作为实施远程精确打击，提高飞行器实际作战效能的关键技术，已经被广泛应用于无人飞行器导航系统中，并且受到世界各国的关注[1]。

蚁群算法（Ant Colony Optimization algorithm，ACO）是由意大利学者 Marco Dorigo 等人在 20 世纪 90 年代初通过模拟自然界中蚂蚁集体觅食的行为而提出的一种基于种群的启发式仿生类算法，最早成功应用于解决 TSP 问题[2~4]。和其他算法相比，蚁群算法具有分布式计算，信息正反馈，鲁棒性强等特点。本文在基本蚁群算法的基础上，提出动态自适应双种群蚁群算法的改进策略，并将改进后的算法应用于无人飞行器航迹规划。

2 航迹规划性能指标的确立

在无人飞行器航迹规划过程中，评价规划质量

* **收稿日期**：2010 年 3 月 5 日，**修回日期**：2010 年 4 月 1 日
作者简介：熊瑜，女，硕士研究生，助教，研究方向：人工智能与知识工程。饶跃东，男，硕士研究生，助教，研究方向：并行分布式及高性能计算。

的好坏需要综合考虑飞行距离,飞行油耗,环境地形,敌方威胁等诸多约束因素。为了能表示航迹规划的综合性能,需要将各方面的约束因素按照某种标准转化为可直接计算比较的量值,然后再确定各个单项指标在综合指标中的权重,最后得到一个表征航迹规划综合代价指标的量值。航迹规划的实质上就是求解每条航迹的综合代价表征值,然后选择出综合代价表征值最优的那条航迹。

2.1 航迹综合代价

为了减小航迹规划空间规模,首先假设无人飞行器在任务执行过程中保持飞行高度和速度不变,并且假设敌方防御区域地形良好,无地形威胁。那么,航迹规划的约束因素就会减少为一个水平航路(二维航路)的优化问题[5]。此时,航迹的综合代价性能指标可表示为最小油耗和最小威胁加权,然后在所经航路上的积分的计算方法,其综合代价函数可表示为:

$$\begin{cases} F = \int_0^L [\lambda\omega_f + \gamma\omega_t]dt \\ \lambda + \gamma = 1 \end{cases} \tag{1}$$

式中,F 表示航迹规划综合性能指标的代价函数,L 表示飞行器所飞行航迹的长度,ω_f表示油耗代价,ω_t表示威胁代价。λ,γ 分别表示油耗代价和威胁代价的权重系数,因这里只考虑这两个约束条件,所以λ,γ 的和为1。它们的大小反映了在进行航迹规划时做出的倾向性选择,若此时侧重考虑油耗代价,则 λ 较大,γ 较小,若侧重考虑雷达等威胁代价,则 λ 较小,γ 较大。在航迹规划节点搜索过程中,对于第 i 条边上的代价函数,可由综合代价函数 F 得出为:

$$\begin{cases} F_i = \lambda\omega_{fi} + \gamma\omega_{ti} \\ \lambda + \gamma = 1 \end{cases} \tag{2}$$

此时,F_i表示每条边上的性能指标代价函数,ω_{fi},ω_{ti}分别表示规划航迹每条边上的的油耗代价和威胁代价。λ,γ 同样表示油耗代价和威胁代价的权重系数,所以和仍然为1。

2.2 油耗代价

由于我们已经假设无人飞行器在飞行过程中保持高度和速度不变,并且不需考虑敌方防御区域的地形威胁,所以油耗代价就应和飞行的航迹长度 L 成正比,或者说 ω_f应是 L 的函数。

$$\begin{cases} \omega_f = \varepsilon \times L \\ \omega_{fi} = \varepsilon \times L_i \end{cases} \tag{3}$$

式中 ε 为油耗代价和航迹长度之间的系数比,L_i为飞行航迹节点间各边的长度。为了简化计算,我们可假设 ε 为1,用飞行航迹的长度代替油耗代价,即 $\omega_f = L$,$\omega_{fi} = L_i$。

2.3 威胁代价

为了简化计算,假定在待规划航迹任务环境下只存在雷达威胁,且各雷达之间互不影响,无人飞行器具有相同的雷达反射截面。则此时飞行器在空间中某个位置 X 所受到的第 j 个雷达的威胁代价只与飞行器与该雷达间的距离有关,其强度与其到雷达的距离的四次方成反比[6]。

$$\omega_{tj}(x) = K_j/(d_j)^4 \tag{4}$$

式中 K_j为反映第 j 个威胁的强度参数,为了简化计算,取 K_j为1,d_j表示无人飞行器到敌方阵地雷达 j 的距离。在实际计算中,第 i 条航迹边威胁代价的计算理论上需要沿第 i 段航迹进行积分。为减少计算量,本文采用只计算航迹段上的若干个点的威胁代价的和。在第 i 条航迹边上取九个等分段点 $L_i/10$、$2L_i/10$、$3L_i/10$、……、$9L_i/10$ 将第 i 条航迹边等分成十段,则第 i 条航迹边威胁代价的近似计算公式如:

$$\omega_{ti} = L_i \sum_{j=1}^{n} [\omega_{tj}(L_i/10) + \omega_{tj}(2L_i/10) + \cdots + \omega_{tj}(9L_i/10)] \tag{5}$$

式中,L_i表示第 i 条航迹边的长度,n 表示已知雷达威胁的个数。根据式(4),$\omega_{tj}(L_i/10)$可表示为 $1/(d_{1/10,i,j})^4$(取式(4)中 K_j为1),即第 i 条边的1/10处的点距离第 j 个威胁点的距离,其余的点以此类推。

3 基本蚁群算法航迹规划应用

将基本蚁群算法应用于航迹规划中时,首先将航迹空间进行网格划分,然后把 m 只蚂蚁放置于航迹规划的起始点,通过状态转移规则,使每只蚂蚁从一个状态(一个节点)转移到另一个状态(另一个节点),直到最终达到目标点,完成一条候选航路,即航迹规划问题的一个可行解。当所有蚂蚁都完成了各自的候选航路选择后,再利用信息素更新规则,将蚂蚁所经过路径上的信息素进行更新。通过更新,引导蚂蚁最终能迭代搜索到航迹规划的最优解。

3.1 航路的生成

如图1所示,假设无人飞行器的起始点为 S,进攻目标点为 G。首先对航迹空间进行网格划分,飞行器按照网格的交点(节点)生成飞行航迹的位

置点。如图所示，除了搜索范围边界的节点（如 S，G）指向与其相邻的其他五个节点外，其它节点均指向相邻的八个节点。所以在搜索过程中，除边界节点外，中间航迹点均可向八个方向搜索转移，即均有八个候选转移节点。根据状态转移规则计算出八个节点的转移概率并转移到下一个节点，直至蚁群到达目标点完成最后搜索并生成候选航路。

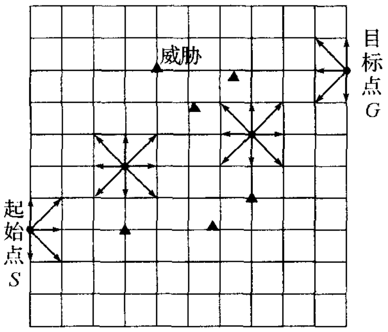

图 1　航迹网格划分图

3.2　状态转移规则

如果蚂蚁当前位于航迹位置点 r，允许转移航迹节点的集合为 $allowed_k$，则从航迹节点 r 转移到航迹点节 s 的状态转移概率公式[7]如下所示：

$$p_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)]^\alpha[\eta(r,s)]^\beta}{\sum\limits_{s\in allowed_k}[\tau(r,s)]^\alpha[\eta(r,s)]^\beta} & \text{if } s\in allowed_k\\ 0 & \text{otherwise}\end{cases}$$

$$\eta(r,s)=\frac{1}{F(r,s)} \tag{6}$$

式中，$P_k(r,s)$ 表示蚂蚁 k 由节点 r 转移到节点 s 的状态转移概率；$\tau(r,s)$ 表示处于节点 r 和 s 之间的航迹边 (r,s) 上的信息素强度，$\eta(r,s)$ 为启发函数，表示节点 s 对节点 r 的能见度。在航迹规划中，$\eta(r,s)$ 的值设定为 $1/F_{(r,s)}$，$F_{(r,s)}$ 表示航迹边 (r,s) 的综合代价；$allowed_k$ 表示允许转移航迹节点的集合，即节点 r 所有相邻可行节点的集合；α，β 作为控制参数用来控制信息素强度和能见度的相对重要性。

3.3　信息素更新规则

当蚂蚁完成一次迭代搜索，找到一条航迹规划可行航迹后，需要对当前所经过路径上的信息素进行一次全局更新。其更新规则如下：

$$\tau(r,s)=(1-\rho)\tau(r,s)+\rho\Delta\tau(r,s)$$

$$\Delta\tau(r,s)=\sum_{k=1}^{m}\Delta\tau^k(r,s)$$

$$\Delta\tau^k(r,s)=\begin{cases}\dfrac{Q}{F_k} & \text{if } (r,s)\in \text{tour done by ant } k\\ 0 & \text{otherwise}\end{cases} \tag{7}$$

式中，ρ 表示信息素挥发系数，F_k 表示蚂蚁 k 选择航路的综合代价，Q 为常数。

同时，为了使算法能够更快更好地搜索到最优解，还需要对最小航路代价的蚂蚁走过的路径进行一次信息素更新，以分配更多的信息素到具有更小威胁代价的航迹边上。其更新规则如下：

$$\tau(r,s)=(1-\rho)\tau(r,s)+\rho\Delta\tau^{best}(r,s)$$

$$\Delta\tau^{best}(r,s)=\begin{cases}\dfrac{Q}{F_{best}} & \text{if } (r,s)\in \text{best tour}\\ 0 & \text{otherwise}\end{cases} \tag{8}$$

式中，F_{best} 表示当前最小的航路代价。

4　改进蚁群算法航迹规划应用

4.1　算法改进思路

基本蚁群算法容易陷入局部最优解，是其较为突出的缺点。产生这一现象的根本原因是局部路径上的信息素过量堆积，造成该路径和其他路径之间的信息素差异加大，从而使得后续的蚂蚁都会集中到这一条路径上，造成一种堵塞和停滞现象，表现在算法求解结果上就是导致算法早熟和局部收敛。因此，如何控制信息素浓度，在算法收敛速度和收敛空间之间找到平衡点就成为蚁群算法优化的关键。

定义：在算法迭代过程中，当求出的最优解出现连续 φ（φ 为常数）代相同时，则认为算法求解陷入可能局部最优。

4.1.1　双种群蚁群搜索

针对单一种群蚁群迭代时，由于信息素正反馈机制容易导致算法早熟这一现象，本文提出在算法运行初始阶段即采用双种群蚁群同时独立搜索，由于不同种群的初始参数环境设置不同，因此迭代过程中各条路径上的信息素分布也不会相同。当算法运行过程中，一旦判断出陷入可能局部最优时，则将两个种群相对应路径上的信息素交换。这样，通过信息素的反复震荡，扩大算法的搜索空间，在一定程度上改善算法的全局性。

4.1.2　双向自适应调整信息素挥发系数

信息素挥发系数 ρ 的大小直接关系各条路径上信息素的大小分布，从而直接影响到蚁群算法的全局搜索能力及其收敛速度。在基本蚁群算法中，ρ 的值是固定的，若 ρ 的值选取不当，则处理问题规模较大时会使那些从来没有被搜索到的路径上的信息量减小接近于 0，这不利于寻找更好的解，容

易使算法陷入局部最优。对此,本文在双种群蚁群搜索的基础上,提出一种双向动态自适应调整信息素挥发系数的算法改进方法。基本思想是:对于两个独立种群的信息素挥发系数值 ρ,其取值范围均设置在(0,1)之间。在算法运行初始阶段,将两个种群的 ρ 值分别一个设置为较小值(如 $\rho=0.1$),一个设置为较大值(如 $\rho=0.9$)。在算法迭代过程中,当出现求解陷入可能局部最优时,则分别自适应地动态调整两个种群的 ρ 值,调整方法是初始阶段 ρ 值设置较小种群的 ρ 值逐渐增大,而初始阶段 ρ 值设置较大种群的 ρ 值逐渐减小。这样,不同的种群通过 ρ 值的变化来扩大各自算法的搜索空间,避免搜索过度集中在某些较优的路径上,以此来寻找更优的路径。而对于不同种群来说,由于初始阶段的 ρ 值设置存在较大差异,而使得迭代过程中不同种群各自路径上的信息素存在较大差异,所以当算法陷入局部最优,两种群对应路径上的信息素交换的时候,才可能发生较大的震荡变化,从而进一步拓展算法的搜索空间,使搜索结果收敛于全局最优解。

因此,当算法迭代过程中出现求解陷入可能局部最优时,不同种群的 ρ 采用式(9)进行自适应调整,设两个种群的挥发系数分别为 ρ_A 和 ρ_B,并且 ρ_A 的初始值设为较小值,ρ_B 的初始值设为较大值,则:

$$\rho_A(k)=\begin{cases}\mu_1 * \rho_A(k-1), & \text{若第 } k-1 \text{ 代出现陷入可能局部最优} \\ \rho_A(k-1), & \text{否则}\end{cases}$$

$$\rho_B(k)=\begin{cases}\mu_2 * \rho_B(k-1), & \text{若第 } k-1 \text{ 代出现陷入可能局部最优} \\ \rho_B(k-1), & \text{否则}\end{cases} \quad (9)$$

式中 k 表示迭代的代数,μ_{1}、$_{2}$分别为两个种群信息素挥发系数的动态改变参数,此时由于已设 ρ_A 初始值为较小值,ρ_B 初始值为较大值,因此 μ_1 应大于1,而 μ_2 应小于1。根据实验结果分析,μ_1 取值在(1,1.5),μ_2 取值在(0.5,1)之间对算法优化较好。

4.2 改进蚁群算法航迹规划算法流程

step1 初始化两个种群参数:α_A、β_A、ρ_A、Q_A、m_A、α_B、β_B、ρ_B、Q_B、m_B、λ、γ、$NC_{\max}$,其中 $\rho_A\in(0,1)$ 且取较小值,$\rho_B\in(0,1)$ 且取较大值;

step2 建立覆盖进攻起始点和目标点的网格图,确定起始点、目标点、威胁点的坐标;

step3 while $NC<NC_{\max}$ 进入以下循环;

step4 将两种群 m_A、m_B 只蚂蚁分别放置于航迹规划起始节点上,根据式(1)~式(5)计算相邻可行航迹节点的 ω_f、ω_t 以及综合代价 F;

step5 两种群分别根据式(6)计算选择概率选择下一个到达节点,完成各自的周游;

step6 分别记录 A、B 两个种群本次迭代的最佳航路;

step7 进入保优函数,比较两个种群本次迭代的最佳航路(最佳综合代价),取其短者(小者)为算法在本轮迭代的最佳航迹;

step8 两种群分别根据式(7)和式(8)更新路径上的信息素;

step9 当算法陷入可能局部最优时,进入种群通信函数,交换 A、B 种群对应路径上的信息素;

step10 当算法陷入可能局部最优时,根据式(9)动态改变 ρ_A 和 ρ_B 的值;

step11 两个种群的禁忌表 $Tabu_A$、$Tabu_B$ 分别清零,$NC=NC+1$;

step12 循环结束;

step13 输出最优航迹。

5 航迹规划任务实例仿真

本文假定无人飞行器的飞行范围为20km×20km的区域,进攻起始点的平面坐标为[1,3](单位:km),攻击目标点的平面坐标为[19,17](单位:km)。同时假设无人飞行器在执行任务过程中保持高度和速度不变,而且不考虑敌方阵地地形因素,将航迹规划问题转换为二维航路问题。假设一共八个敌方威胁源(主要考虑雷达威胁),其平面坐标如表1所示。

表1 雷达威胁的平面坐标方位表

起始点坐标	[1,3]	目标点坐标	[19,17]
威胁点1坐标	[5,9]	威胁点5坐标	[10,10]
威胁点2坐标	[11,14]	威胁点6坐标	[15,8]
威胁点3坐标	[9,6]	威胁点7坐标	[14,17]
威胁点4坐标	[16,3]	威胁点8坐标	[19,12]

本文采用MATLAB作为仿真工具,分别采用基本蚁群算法和改进蚁群算法对上述航迹规划任务进行仿真实验。对于基本蚁群算法的实验参数设定如下所示:蚂蚁总数 $m=20$,$\alpha=1$,$\beta=2$,$\rho=0.2$,$\lambda=0.1$,$\gamma=0.9$,$NC_{\max}=100$。为了使两种算法实验结果具有可对比性,对于改进的自适应双种群蚁群算法中的 λ 和 γ 值设定成和基本蚁群算法相同,并且其中一个种群的 α 和 β 值也设定成和基本算法相同,其余参数初始化如表2所示。

表2 改进算法实验参数

α_A	β_A	ρ_A	m_A	α_B	β_B	ρ_B	m_B	μ_1	μ_2	λ	γ
1	2	0.1	20	2	3	0.9	20	1.1	0.9	0.1	0.9

(下转第146页)

[6] 敬忠良,肖刚,等. 图像融合—理论与应用[M]. 北京:高等教育出版社,2007
[7] 曾基兵,陈怀新,王卫星. 基于改进局部梯度的小波图像融合方法[J]. 电视技术,2007,31(8):18～20
[8] 苗启广,王宝树. 基于小波变换与局部能量的多聚焦图像融合[J]. 计算机科学,2005,32(2):229～232
[9] 周锐锐,等. 一种基于局部能量的图像融合方法[J]. 中国体视学与图像分析,2006,11(3):229～229
[10] 刘丽. 多源遥感图像像素级融合方法研究[D]. 长沙:湖南大学,2005

(上接第 44 页)

分别使用 MATLAB 工具根据两种算法进行仿真编程[8～10],同时根据初始化的参数对上述航迹规划任务连续进行五次仿真实验。记录每次实验得到的最小综合代价,其结果对比如表 3 所示。

表 3　仿真实验结果对比表

实验序号	基本蚁群算法航迹综合代价	改进蚁群算法航迹综合代价
1	29.1236	23.6506
2	29.7505	26.6136
3	32.5157	20.8770
4	36.8519	19.0082
5	35.8638	25.9319
最小综合代价	29.1236	19.0082
平均综合代价	32.8211	23.2163

通过仿真结果对比可以看出,在五次实验中,改进蚁群算法不管是最小综合代价,还是平均综合代价上都要优于基本蚁群算法。因此,可以看出,改进算法不仅能够搜索到更优的解,而且在稳定性上也得到了增强,确实能够有效应用于航迹规划并且提高航迹规划的性能指标。

图 2 和图 3 分别为两种算法在五次仿真实验中最优综合代价的航迹仿真图,直接由 MATLAB 编程仿真而得。

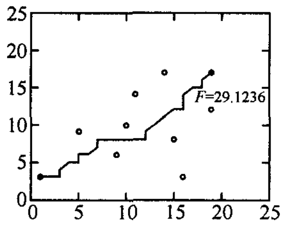

图 2　基本蚁群算法最优航迹图

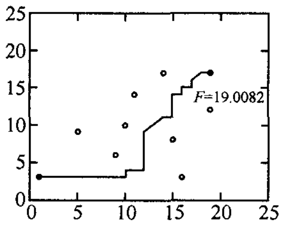

图 3　改进蚁群算法最优航迹图

6　结语

本文在基本蚁群算法的基础上,提出双种群同时搜索,并且双向动态自适应调整信息素挥发系数的算法改进策略,在一定程度上提高了算法搜索的全局性和稳定性。将改进后的算法应用于无人飞行器航迹规划,经过仿真实验证明此算法改进能够有效提高航迹规划的质量。在实验中我们发现,不同种群参数的设置对航迹规划的结果影响较大。因此,下一步工作将继续对参数配比加以研究。

参考文献

[1] 闵昌万,袁建平. 军用飞行器航迹规划综述[J]. 飞行力学,1998,16(4):14～19
[2] Colorni A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies[C]//Proceedings of the 1st European Conference on Artificial Life,1991:134～142
[3] Dorigo M. Optimization, learning and natural algorithms[D]. Ph. D. Thesis,Department of Electronics, Politecnico diMilano, Italy,1992
[4] Dorigo M, Maniezzo V, Coloni A. Antsystem:optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cyberneties-PartB,1996,26(1):29～41
[5] Timothy W M, Randal W B. Trajectory planning for coordinated rendezvous of unmanned air vehicles. AIAA-2000-4339-CP,2000
[6] Kevin B Judd, Timothy W Mclain. Spline based path planning for unmanned air vechicles[J/OL]. AIAA 2001-4238. 2001
[7] 段海滨. 蚁群算法原理及应用[M]. 北京:科学出版社,2005,12
[8] 胡小兵,袁锐,黄席樾,等. 蚁群算法原理的仿真研究[J]. 计算机仿真,2004,21(8):125～128
[9] 周伟,刘粉林,吴灏,等. 简单蚁群算法的仿真分析[J]. 控制与决策,2003,18(3):317～319
[10] 孙祥,徐流美,吴清. MATLAB7.0 基础教程[M]. 北京:清华大学出版社,2005,5